Point Cloud Registration Algorithm Based on Entropy Criterion Genetic Algorithm Postprint

Authors: Chen Jie, Cai Yong, Zhang Jiansheng

Date: 2018-05-20T00:00:00+00:00

Abstract

The Iterative Closest Point (ICP) algorithm is typically employed for fine registration of point clouds due to its high registration accuracy; however, its registration accuracy and iterative convergence depend heavily on the initial position of the point clouds to be registered. This paper proposes an algorithm for solving the spatial optimal transformation matrix that integrates genetic algorithms with spatial distribution entropy, utilizing a novel evaluation method for point cloud spatial position—spatial distribution entropy—as the objective function of the genetic algorithm. Genetic operators are employed to guide the search direction of solutions, and through continuous iteration of new populations, the spatial distribution entropy is minimized. Upon completion, the optimal individual is decoded to achieve coarse registration of point clouds. Experimental results demonstrate that the proposed algorithm is effective and feasible, overcoming the limitation of traditional methods that fail to provide satisfactory initial alignment positions in the presence of point cloud defects and noise points. Within the allowable error range, direct point cloud stitching can be achieved.

Full Text

Point Cloud Registration Based on Entropy Criterion Genetic Algorithm

Chen Jie[1] , Cai Yong[1] , Zhang Jiansheng[1] [1]Southwest University of Science and Technology a. School of Manufacturing Science & Engineering b. Key Laboratory of Testing Technology for Manufacturing Process, Ministry of Education Mianyang, Sichuan 621010, China

Abstract: The Iterative Closest Point (ICP) algorithm is widely used for fine registration of point clouds due to its high accuracy, but its registration precision and convergence depend heavily on the initial position of the point clouds to be registered. This paper proposes a spatial optimal transformation matrix

solution algorithm that integrates genetic algorithms with spatial distribution entropy. Using a novel point cloud spatial position evaluation method—spatial distribution entropy—as the objective function of the genetic algorithm, the algorithm employs genetic operators to guide the search direction. Through continuous iteration of new populations until the spatial distribution entropy is minimized, coarse registration of point clouds is achieved by decoding the optimal individual. Experiments demonstrate that the proposed algorithm is effective and feasible, overcoming the limitation of traditional methods that fail to provide good initial alignment positions when point clouds contain defects and noise points. Within acceptable error bounds, the algorithm can directly achieve point cloud registration.

Keywords: iterative closest point algorithm; genetic algorithm; spatial distribution entropy; registration

0 Introduction

As 3D scanners continue to advance toward higher precision and quality and become increasingly widespread, point cloud registration technology has emerged as a crucial technique with extensive applications across numerous industries, including reverse engineering, product inspection, robot pose adjustment, cultural heritage preservation, computer vision, computer graphics, and pattern recognition. This makes it a highly important research area. When using various scanning devices to accurately sample object contours, we obtain discrete points in three-dimensional space, and the collection of all these points is called a point cloud. These sampled points representing model contours enable 3D reconstruction in computer systems. For large or geometrically complex objects, each scan can only capture a portion of the surface, requiring multiple scans to create a complete model. The resulting point clouds are initially unrelated to each other, and to represent the complete object contour accurately, point clouds from multiple viewpoints must be transformed into a common coordinate system through coordinate transformation—this operation is point cloud registration.

Point cloud registration essentially involves computing rotation and translation matrices to align points from the same physical location on an object's surface. A common and effective approach is to perform coarse registration followed by fine registration. Coarse registration places two point clouds with arbitrary initial positions into approximately correct relative positions, providing a good initial state for fine registration to converge to the global minimum and improving both efficiency and accuracy. Fine registration then achieves complete alignment of corresponding points on the object surface.

Besl et al. first proposed the Iterative Closest Point (ICP) algorithm in 1992, which remains the most mature and effective fine registration algorithm when applied correctly. The fundamental principle of the original ICP algorithm is

to find corresponding point pairs between two point sets, compute the mean squared Euclidean distance between all correspondences, update the relative position between point clouds, and repeat this process until convergence—when the mean distance is minimized and the two point clouds reach optimal alignment.

However, the classic ICP algorithm has several limitations. First, its convergence speed and whether it converges to the correct position depend heavily on the initial alignment. Second, each iteration requires finding nearest neighbors for every point, which becomes extremely time-consuming for large point clouds. Third, since point clouds typically overlap only partially and contain noise, some points lack true correspondences. The original ICP algorithm assumes such points do not exist, and when they are present but not removed, they significantly degrade performance. To address these limitations, numerous scholars have proposed improved variants of the ICP algorithm.

To overcome ICP' s dependency on initial position, researchers have developed various coarse registration methods. Reference [2] employs Principal Component Analysis (PCA) for fast registration, but it cannot provide good initial positions for point clouds with substantial noise or partial overlap. Reference [3] extracts boundary feature points for coarse registration, but this method suffers from low precision and time-consuming feature extraction. Reference [4] uses an iterative minimum spatial distribution entropy method for coarse registration, yet this approach also exhibits low precision and high computational cost.

In recent years, several stochastic algorithms have been proposed. Reference [5] combines genetic algorithms with Mean Squared Error (MSE) metrics for coarse registration, achieving high accuracy but low efficiency. Reference [6] integrates genetic algorithms with O' Rourke' s algorithm to solve for minimum bounding boxes. Chen et al. [8] first applied the RANSAC algorithm to 3D point cloud registration in 1999, demonstrating high robustness without requiring geometric features or initial positions, and proving effective for minimally overlapping point clouds, though with substantial time complexity. Reference [9] introduced the 4PCS (4-Point Congruent Sets) technique combined with RANSAC, replacing random sampling of three points with four-point sets that better exploit geometric relationships to accelerate correspondence search. While 4PCS significantly improves RANSAC efficiency, its computational complexity remains high for massive datasets. Reference [10] employs a sample-sphere registration method that accelerates the search for three-point bases while offering greater resistance to noise and outliers than previous RANSAC methods. Reference [12] addresses models with large local deformations by using genetic algorithms to provide initial positions, then applying ICP to compute deviation values, setting thresholds to remove points with large deformations, and finally performing registration using least squares. Reference [13] improves the genetic algorithm' s objective function to the sum of squared distances from measurement points to triangular facets, using reference spheres to accelerate distance calculations and save considerable time.

Based on this analysis, we propose a coarse registration method based on entropy criterion genetic algorithms, using spatial distribution entropy as the objective function. This approach can quickly place two point clouds with no prior information into approximate alignment, providing good initial values for fine registration. Spatial distribution entropy accurately measures the alignment degree of point clouds, granting the algorithm excellent robustness.

---

1 Algorithm Overview

The algorithm first constructs a three-dimensional cubic grid for both the source point cloud P and target point cloud Q using an octree, then counts the number of points within each cubic cell, and finally computes the point cloud spatial distribution entropy. Point cloud registration proceeds in two stages: coarse registration and fine registration. Coarse registration employs spatial distribution entropy as the objective function, using genetic algorithms to solve for the optimal spatial transformation matrix. Fine registration uses an improved ICP algorithm to achieve precise alignment.

1.1 Spatial Voxel Grid

This algorithm utilizes an octree [14] to establish spatial cubic grids, with the termination condition for subdivision being the squared diagonal length of a node grid:

$$l^2 = (x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2 + (z_{\max} - z_{\min})^2$$

where $x_{\max}, y_{\max}, z_{\max}$ represent the maximum values along each coordinate axis of the node grid, and $x_{\min}, y_{\min}, z_{\min}$ represent the minimum values.

All grids have identical sizes. Since point cloud data is typically non-uniform, the number of points in each grid varies after voxelization. Overlapping regions of point clouds are relatively denser than other areas, and this density variation is well-described by the point count in each grid—denser point clouds correspond to grids containing more points, while sparser regions contain fewer.

1.2 Spatial Distribution Entropy

The spatial distribution entropy [4] proposed in this paper describes point cloud positional information and shares similarities with information entropy [15]. Point clouds scanned from different viewpoints typically have arbitrary spatial positions with significant positional differences and high uncertainty. When two point clouds are successfully registered, their relative position becomes minimal and uniquely determined. Spatial distribution entropy accurately characterizes this positional relationship during registration.

The computation process for point cloud spatial distribution entropy is as follows:

1. Point cloud D contains N data points, which are uniformly voxelized into a 3D grid using an octree.
2. After grid partitioning, record the number of points assigned to each cubic cell as n_{ijk}, and compute the probability of points falling into each cell: $p_{ijk} = \frac{n_{ijk}}{N}$, where i, j, k represent the grid indices along each coordinate axis, with $i = 1, 2, 3, \ldots, n_x$, $j = 1, 2, 3, \ldots, n_y$, $k = 1, 2, 3, \ldots, n_z$, and n_x, n_y, n_z denote the number of grids along each axis.
3. Compute the Spatial Distribution Entropy (SDE) of point cloud D:

$$\mathrm{SDE}(D) = -\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\sum_{k=1}^{n_z} p_{ijk}\ln(p_{ijk})$$

The algorithm demonstrates that computing point cloud spatial distribution entropy (SDE) only requires establishing a 3D grid and counting points in each cell, resulting in low time complexity. In contrast, Mean Squared Error (MSE) [4] requires finding nearest neighbors for every point in one point cloud to the other, performing a global search each time, leading to high computational complexity.

Given two perfectly aligned bunny model point clouds P and Q, we perform the following operation on Q: rotate it clockwise by angle θ around the Z-axis, with θ taking values at 0.0314 rad intervals in the range $[-\pi, \pi]$. The rotated Q is denoted as $Q(\theta)$. The combined point cloud $D(\theta) = P \cup Q(\theta)$ is used to compute both MSE and SDE. Figure 1(a) shows MSE values, while Figure 1(b) shows SDE values. Table 1 compares the time required for MSE and SDE to evaluate point cloud spatial positions.

[Figure 1: see original paper] (a) MSE (b) SDE

Table 1 Time comparison between MSE and SDE for evaluating point cloud spatial positions (seconds)

As shown in Figure 1, both MSE and SDE satisfy spatial position estimation requirements, with both achieving global minima when point clouds are perfectly aligned. Table 1 reveals that SDE estimates point cloud spatial positions far more efficiently than MSE, making it suitable for frequent objective function evaluations.

Since SDE is based on the probability distribution of points within cubic grids, finer grid partitioning yields more precise spatial position representation. However, computation time increases with the number of grids, making the termination condition crucial for algorithm performance. Considering the effectiveness and efficiency of SDE for spatial position estimation, this paper selects a node grid diagonal squared length of 0.003 for the bunny model. Table 2 compares runtime for different node grid sizes.

Table 2 Comparison of runtime for different node grid sizes (seconds)

1.3 Genetic Algorithm

Genetic algorithms [16,17] are evolutionary optimization algorithms. Through evolutionary learning that preserves superior individuals while generating new ones, genetic algorithms exhibit strong robustness and enable multi-point search in the design space, providing powerful global search capabilities. The method features global convergence properties and does not require strict initial values. The genetic algorithm process is described as follows:

a) **Encoding**: Convert the problem solution into a genotypic string structure.
b) **Initial population generation**: Randomly generate a population of genotypic data.
c) **Fitness evaluation**: Determine appropriate fitness functions to evaluate individual adaptability based on objectives.
d) **Selection and inheritance**: Superior individuals in the population are selected for inheritance to the next generation, with offspring produced through crossover and mutation of parents to generate even better individuals.
e) **Termination**: When termination conditions are met, the algorithm stops selection and inheritance, with the optimal individual in the evolutionary process being the optimal solution.

---

2 Registration Based on Entropy Criterion Genetic Algorithm

This paper employs an entropy criterion-based genetic algorithm for coarse registration. First, the objective function is established, where spatial distribution entropy measures the alignment degree of overlapping regions to determine positional relationships. Then, genetic algorithm operations—selection, reproduction, crossover, and mutation—are iteratively applied. Finally, the optimal rotation and translation matrices are identified.

2.1 Objective Function Establishment

The registration objective is to transform point clouds scanned from different viewpoints into a common coordinate system. Point cloud spatial distribution entropy can rapidly and accurately evaluate transformation results, making it suitable as the genetic algorithm' s objective function. The problem is thus formulated as minimizing the spatial distribution entropy. Let the spatial distribution entropy function be $f(T) = \mathrm{SDE}(P, T(Q))$, where P is the reference point cloud and $T(Q)$ is the transformed target point cloud. This function can precisely obtain local optimal solutions. Since each individual in the genetic

algorithm requires objective function evaluation, using SDE ensures solution accuracy, effectively narrows the search space, and improves algorithm efficiency.

2.2 Genetic Algorithm Selection

2.2.1 Chromosome Real Number Encoding Objects in 3D space have six degrees of freedom: three rotations and three translations. The transformation matrix can be represented by six variables: rotation variables $[\alpha, \beta, \gamma]$ with range $[-\pi, \pi]$, and translation variables $[T_x, T_y, T_z]$ with ranges determined from reference point cloud P: $[x_{\min} - x_{\max}, x_{\max} - x_{\min}]$, $[y_{\min} - y_{\max}, y_{\max} - y_{\min}]$, $[z_{\min} - z_{\max}, z_{\max} - z_{\min}]$, where $x_{\max}, y_{\max}, z_{\max}$ and $x_{\min}, y_{\min}, z_{\min}$ are the maximum and minimum values of point cloud P along each coordinate axis.

2.2.2 Fitness Function In genetic algorithms, the fitness function is the most important basis for "natural selection." This paper' s objective function is spatial distribution entropy (SDE), where individuals with smaller SDE values are superior and have higher selection probabilities. This algorithm employs rank-based fitness assignment. Individuals are sorted by their SDE values in ascending order, and fitness values (FitnV) are calculated accordingly. Larger FitnV values indicate superior individuals.

The fitness function is:

$$\text{FitnV} = \frac{1}{\sqrt{sp}} \left[2 - \text{Pos} + 2(\text{Pos} - 1) \times (sp - 1)\right]$$

where sp is the individual' s SDE value, Pos is the individual' s rank position in the population, and N_{ind} is the population size.

2.2.3 Selection and Inheritance This paper adopts the elitist strategy for selection operations, where the current best individual is directly passed to the next generation. Random universal sampling is used to select other chromosomes, where selection probability is proportional to the fitness function.

Since chromosomes use real number encoding, the crossover operator employs discrete recombination from real-valued crossover operators. The mutation operator uses real-valued population mutation. Mutation for a variable is computed as:

$$X' = X + \text{MutMx} \times \text{range} \times \beta \times \text{delta}$$

where MutMx controls whether mutation occurs, taking values randomly from $\{1, 0, -1\}$; range $= 0.5 \times$ variable domain (see Section 2.2.1); β is a compression factor for mutation range, a value between $[0, 1]$; and delta is the mutation step size.

2.3 Algorithm Steps

This paper' s genetic algorithm implementation utilizes the popular and well-developed MATLAB Genetic Algorithm Toolbox [18]. With the objective function and selection/inheritance operations defined above, the algorithm proceeds as follows:

1. Initialize population P within maximum variable domains.
2. Apply population P to the target point cloud to find the x, y, z maxima and minima for each initial transformation, then determine the overall spatial x, y, z maxima and minima to define the maximum transformation space.
3. Compute objective function values and fitness values.
4. Select the best individual in the population to obtain its spatial distribution entropy SDE, set the global minimum $\mathrm{SDE}_{\min} = \mathrm{SDE}$, and set the maximum generation G.
5. Perform selection of superior individuals based on fitness values, then apply crossover and mutation to produce a new evolved population.
6. Compute fitness values FitnV for all individuals in the new generation, compare the best individual' s SDE with the global minimum $\mathrm{SDE}_{\min}$, and update $\mathrm{SDE}_{\min}$ if $\mathrm{SDE} < \mathrm{SDE}_{\min}$.
7. Check if the current generation reaches the maximum G or if SDE has converged. If not, return to step (5); otherwise, output the global optimal individual, decode to obtain the optimal rotation and translation matrices, and terminate.
8. Update point cloud positions and output the registered point clouds.

The entropy criterion genetic algorithm flowchart is shown in Figure 2.

[Figure 2: see original paper] Flowchart of entropy criterion genetic algorithm

---

3 Experimental Results and Analysis

To verify the algorithm' s feasibility, robustness, and effectiveness, coarse registration experiments using the entropy criterion genetic algorithm were conducted on MATLAB on an Intel Core-i5 with 2 GB RAM running Windows 7. Fine registration using the improved ICP algorithm was implemented in Visual C++. Verification focused on two aspects: point clouds with defects and point clouds with substantial noise. Using Stanford' s bunny, cow, and man models as examples, data from two different viewpoints (P and Q) were processed.

Figures 3-6 show the relative positions of reference and target point clouds before and after coarse registration, where (a) shows the initial 90° misalignment and (b) shows the coarse registration results.

[Figure 3: see original paper] Relative positions of simplified bunny point clouds before and after coarse registration (a) 90° misalignment (b) After coarse regis-

tration

[Figure 4: see original paper] Relative positions of noisy point clouds before and after coarse registration (a) 90° misalignment (b) After coarse registration

[Figure 5: see original paper] Relative positions of point clouds with 1/2 data missing before and after coarse registration (a) 90° misalignment (b) After coarse registration

[Figure 6: see original paper] Relative positions of point clouds with 3/4 data missing before and after coarse registration (a) 90° misalignment (b) After coarse registration

To observe the data variation patterns across generations, Figure 7 shows the evolution of the minimum SDE and mean SDE per generation for the simplified bunny model from Figure 3.

[Figure 7: see original paper] Objective function values and performance tracking during optimization

Figure 7 demonstrates that the mean SDE decreases across generations, with the population evolving toward the optimal solution. The global optimum updates during iteration based on local optima, eventually converging to the global minimum SDE. Figures 3-6(b) all represent results converged to the global minimum SDE. The registration results demonstrate that the proposed algorithm can handle models with small overlapping regions while resisting interference from noise and outliers.

A comparison of coarse registration effects between the proposed algorithm, MSE-based genetic algorithm, reference sphere-accelerated genetic algorithm [13], fast PCA, and 4PCS is shown in Figure 8.

[Figure 8: see original paper] Relative positions after coarse registration using different methods (a) Before registration (b) MSE (c) Reference [13] (d) PCA (e) 4PCS (f) Proposed algorithm

Figure 8 shows that the proposed algorithm, MSE, reference [13], and 4PCS can all provide initial positions for complete and partially overlapping point clouds, while PCA fails for partially overlapping cases. The results demonstrate that the entropy criterion genetic algorithm effectively performs coarse registration.

Table 3 Coarse registration time for different models (seconds)

The runtime of genetic algorithms and RANSAC is highly dependent on parameter settings. All three algorithms were configured with an initial population of 80 and 40 generations. Figure 9 shows registration time versus point cloud size for the proposed SDE method, MSE, and reference [13].

[Figure 9: see original paper] Point cloud size versus time curve

Figure 9 indicates that registration times for reference [13] and MSE increase rapidly with point count, while the SDE method shows a gentle increase, demon-

strating that the entropy criterion genetic algorithm can handle large-scale point cloud registration. Both MSE and reference [13] require global searches for corresponding points to compute distances, causing search time to increase dramatically with point count.

Using the initial position provided by the entropy criterion genetic algorithm for coarse registration, we then applied an improved ICP algorithm using KD-trees for nearest neighbor search for fine registration, comparing it with traditional ICP [1] and reference [3]. The results are shown in Table 4.

Table 4 Comparison of point cloud registration performance

Registration accuracy evaluation adopts the standard from reference [3], calculated as:

$$\text{Fail}(p_i^1, p_i^2) = \begin{cases} 1 & \text{if } d(p_i^1, p_i^2) > \delta \\ 0 & \text{if } d(p_i^1, p_i^2) \leq \delta \end{cases}$$

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \text{Fail}(p_i^1, p_i^2)$$

where $d(p_i^1, p_i^2)$ represents the distance between corresponding points, δ is the distance threshold, and $\text{Fail}(p_i^1, p_i^2)$ indicates whether the correspondence meets accuracy requirements.

---

4 Conclusion

Point cloud registration is fundamental to reverse engineering, where robustness, efficiency, and accuracy are critical. The proposed algorithm demonstrates good performance for raw point clouds with outliers and partial overlap, though its efficiency requires further improvement. The algorithm features:

a) By avoiding exhaustive search of the entire variable domain through genetic algorithms, the evolutionary guidance enables rapid acquisition of near-optimal solutions, effectively improving the efficiency of minimizing point cloud spatial distribution entropy.

b) Spatial distribution entropy is computed based on point cloud density, making spatial position estimation relatively insensitive to outlier points.

c) The parallel computation capability of genetic algorithms can be further exploited to improve solution efficiency.

The ICP algorithm' s high accuracy makes it the standard for fine registration, making the challenge of providing good initial positions quickly and accurately particularly important. Future work will focus on improving the proposed algorithm to further enhance its efficiency.

References

[1] Besl P J, Mckay H D. A method for registration of 3-D shapes [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2002, 14(2): 239-256.

[2] Dai Jinglan, Chen Zhiyang, Ye Xiuzi. ICP algorithm in point cloud registration [J]. Journal of Image and Graphics, 2007, 12(3): 517-521.

[3] Wang Xin, Zhang Mingming, Yu Xiao, et al. Point cloud data registration using improved iterative closest point method [J]. Optics and Precision Engineering, 2012, 20(9): 2068-2077.

[4] Zuo Chao, Lu Min, Tan Zhiguo, et al. A new point cloud stitching algorithm [J]. Chinese Journal of Lasers, 2012, 39(12): 211-218.

[5] Silva L, Bellon O R P, Boyer K L. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2005, 27(5): 758-763.

[6] Sun Dianzhu, Shi Yang, Liu Huadong, et al. Solution of minimum bounding box for scattered point cloud based on genetic algorithm [J]. Journal of Beijing University of Aeronautics and Astronautics, 2013, 39(8): 995-998.

[7] Mavridis P, Andreadis A, Papaioannou G. Efficient sparse ICP [J]. Computer Aided Geometric Design, 2015, 35-36: 16-26.

[8] Chen C S, Hung Y P, Cheng J B. RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2002, 21(11): 1229-1234.

[9] Aiger D, Mitra N J, Cohen-Or D. 4-points congruent sets for robust pairwise surface registration [J]. ACM Trans on Graphics, 2008, 27(3): 85.

[10] Meng Y, Zhang H. Registration of point clouds using sample-sphere and adaptive distance restriction [J]. Visual Computer, 2011, 27(6-8): 543-553.

[11] Lei Yuzhen, Li Zhongwei, Zhong Kai, et al. Mismatched marker correction method based on random sample consensus algorithm [J]. Acta Optica Sinica, 2013, 33(3): 205-212.

[12] Wu Dianliang, Huang Hailiang, Ding Yucheng, et al. Surface matching based on genetic algorithm and least squares method [J]. Acta Aeronautica et Astronautica Sinica, 2002, 23(3): 285-288.

[13] Ping Xueliang, Geng Lu, Hua Ting, et al. Application of genetic algorithm in point cloud registration technology [J]. Mechanical Science and Technology, 2010, 29(6): 809-812.

[14] Zheng Dejiong, Lu Keqing. Research on fast 3D point cloud picking method based on adaptive octree [J]. Mechanical & Electrical Engineering, 2016, 33(4): 417-420.

[15] Lee C Y, Han S K. Evolutionary optimization algorithm by entropic sampling [J]. Physical Review E, 1998, 57(3): 3611-3617.

[16] Goldberg D E. Genetic algorithms in search, optimization, and machine learning [M]. 1989.

[17] Qu Zhijian, Zhang Xianwei, Cao Yanfeng, et al. Research on genetic algorithm based on adaptive mechanism [J]. Computer Application Research, 2015, 32(11): 3222-3225, 3229.

[18] Lei Yingjie, Zhang Shanwen, Li Xuwu, et al. MATLAB genetic algorithm toolbox and its applications [M]. Xi' an: Xidian University Press, 2005: 8-14.

Note: Figure translations are in progress. See original paper for figures.